Summer 2008

CS 480/680: Comparative Languages

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

CS 480/680 Comparative Languages

- **Instructor** : T. K. Prasad
- **Phone No.** : (937)-775-5109
- **Email** : t.k.prasad@wright.edu
- **Home page**: http://www.cs.wright.edu/~tkprasad/

- **Quarter** : Summer, 2008
- **Class Hrs** : MW, 6:05 - 7:20pm, Health Sci Bldg 134.
- **Office Hrs** : MW, 5:00 - 6:00pm, 395 JC (or by appt.)

Course Description

This course will introduce fundamental concepts and paradigms underlying the design of modern programming languages. For concreteness, we study the details of an object-oriented language (e.g. `Java`), and a functional language (e.g., `Scheme`). The overall goal is to enable comparison and evaluation of existing languages. The programming assignments will be coded in `Java 5` and in `Scheme`.

Prerequisites

- Data Structures and Algorithms. (Equivalently, CS400/600.)
- Experience with programming in imperative languages such as C/C++, Pascal, or Ada.

Course Text and Material

1. On-line Lecture Notes.
2. K. Arnold, J. Gosling, and D. Holmes: The Java Programming Language. Addison-Wesley Publishing Co., 4th Edition, 2005. ISBN **0-321-34980-6**

References

1. Michael L. Scott, Programming Language Pragmatics. Morgan Kaufmann Publishers, 2nd Edition, 2006. ISBN 0126339511
2. The Java Tutorial
3. Ravi Sethi, Programming Languages: Concepts and Constructs. Addison-Wesley Publishing Co., 2nd Edition, 1996. ISBN 0-201-59065-4
4. R. Kent Dybvig, The Scheme Programming Language, 3rd Edition. Prentice Hall, 2003.
5. Scheme : Language Reference Manual
6. Chez Scheme Download Site (http://www.scheme.com)
7. DrScheme Download Site (http://www.drscheme.org/)

8. Jython Home Page
9. Dive into Python

Relevant Websites

- Sun's Java Page
  - Java 5.0 Core APIs
- The Teaching About Programming Languages Project

Download Sites

1. JDK Download (http://java.sun.com/javase/downloads/index.jsp)
2. Eclipse Download (http://www.eclipse.org/downloads/index.php)
3. TextPad Editor (www.textpad.com)

Java IDE Tutorials by Y. Daniel Liang

1. Compiling and Running Java from the Command Window
2. Compiling and Running Java from TexPad
3. NetBeans Tutorial
4. Eclipse Tutorial

Course Load

The course load includes a mix of homeworks and programming assignments worth 30 points, a midterm worth 30 points and a final worth 40 points. Normally, CS680 students are assigned additional homework problems and are expected to solve additional/different problems in the tests.

Grading

The letter grades will be assigned using the following scale: A[90-100], B[80-90), C[70-80), D[60-70), and F[0-60). However, I reserve the right to adjust the scale somewhat to utilize the gaps in the distribution. Academic dishonesty will be "rewarded" with a grade of "F". "Sharing/reuse" of solutions to assignment problems is strictly prohibited.

Attendance Policy

All registered students are expected to attend all lectures. In case a student is absent from a lecture due to unavoidable circumstances, the student is still responsible for the material covered in the class, as it is typically available from the course web-page well in advance. Furthermore, the student is expected to find out about in-class announcements from their colleagues/instructor.

Class Schedule and Syllabus

	Topic
Class 1	Evolution of Programming Languages
Class 2	Syntax Specification : Grammars
Class 3	Object-Oriented Programming
Class 4	Symbolic Data; List Processing
Class 5	Styles : Functional vs Procedural
Class 6	Recursive Definitions (Scheme-Startup)(Examples)
Class 7	Abstraction : Higher Order Functions
Class 8	Scoping; Closures
Class 9	**Midterm (July 16)**
Class 10	Java Design Goals
Class 11	Types, Values, Variables
Class 12	Arrays; Classes
Class 13	Inheritance; Polymorphism
Class 14	Interfaces; Packages; Strings
Class 15	Exceptions
Class 16	Threads
Class 17	(continue) (Scripting vs Systems PL)
Class 18	Scheme Interpreter
Class 19	(continue)
Class 20	(continue)
Class *	Parameter Passing Mechanisms
Class *	Implementing Subprograms
	Final (August 20, 5:45pm-7:45pm)

Assignments (Summer 2008)

- Assignment 1
- Assignment 2

Exams (Spring 2008)

- Midterm
- Final

T. K. Prasad (06/09/08 10:47:16 AM)

CS 765
Foundations of Neurocomputation

Instructor: Dr. M. M.Rizki
Office: 432 Russ Engineering
Phone: 775-5117
Email: mateen.rizki@wright.edu

Office Hours: Tuesday and Thursday 5:30-6:30 PM and by appointment

Course Objectives: This course is designed to help you develop a solid understanding of neural network algorithms and architectures. At the end of this course you should be able to read and critically evaluate most neural network papers published in major journals, (e.g. IEEE Transaction on Neural Networks, Neural Networks, and Neural Computation). In addition, you should be able to implement a broad range of network architectures and learning algorithms for a variety of applications.

Prerequisites: Familiarity with multivariate calculus, linear algebra and matrix algebra.

Familiarity with algorithmic complexity concepts and programming.

Textbooks:

Required: Neural Networks A Comprehensive Foundation by Simon Haykin,Prentice-Hall, 1999

Recommended: If you are not familiar with Matlab, you may want to obtain a book on programming in Matlab.

Workload:

2-3 Programming / Homework Exercises	30%
1 Course Project / Presentation	30%
1 Midterm Examination	20%
1 Final Examination	20%

Topics:

Introduction to artificial neural networks	Ch. 1
Overview of principles and methods of neural computing	Ch. 2
Single layer networks	Ch. 3
Multilayer networks	Ch. 4
Radial-Basis function networks	Ch. 5
Self-organizing maps and vector quantization	Ch. 9
Neurodynamics	Ch. 14
Recurrent networks	Ch. 15
Applications of Neural Networks	
Project Presentations	